In the Claims

Please replace all prior versions, and listings, of claims in the application with the following list of claims:

1. (Currently amended) A digital signal processor comprising:

an address generator configured to generate speculative data addresses in response to an address operand and one or more address parameters;

a pipelined execution unit configured to execute instructions in an instruction pipeline ~~having~~ including each of a plural number of stages required to execute the instructions from an initial stage in which instructions are fetched through a final stage during which execution of the instructions are completed, the plural number of stages using data at locations specified by the speculative data addresses;

a speculative register file capable of simultaneously storing at least a number of speculative data addresses equal to or greater than the plural number of pipeline stages and at least one architectural data address, the speculative register file configured to hold the speculative data addresses as corresponding instructions advance through the instruction pipeline and at least one speculative data address after the at least one speculative data address becomes a respective architectural data address; and

control logic configured to write speculative data addresses to the speculative register file as the speculative data addresses are generated by the address generator and to supply speculative data addresses and architectural data addresses to the address generator.

2. (Canceled).

3. (Previously presented) A digital signal processor as defined in claim 1, further comprising an architectural register file configured to hold architectural data addresses, wherein the control logic is configured to move architectural data addresses from the speculative register file to the architectural register file in the event of a conflict for use of the speculative register file.

4. (Original) A digital signal processor as defined in claim 3, wherein the control logic is configured to write speculative data addresses to successive slots in the speculative register file.

5. (Original) A digital signal processor as defined in claim 4, wherein the control logic is configured to increment a pointer to a next available slot in the speculative register file.

6. (Original) A digital signal processor as defined in claim 5, wherein the control logic is configured to wrap the pointer from an end of the speculative register file to a start of the speculative register file.

7. (Original) A digital signal processor as defined in claim 3, wherein the control logic is configured to mark as architectural an entry in the speculative register file in response to the corresponding instruction being completed by the pipelined execution unit.

8. (Original) A digital signal processor as defined in claim 7, wherein the control logic is configured to mark as empty a slot in the speculative register file containing an old architectural data address when a current architectural data address is defined.

9. (Original) A digital signal processor as defined in claim 7, wherein the control logic is configured to mark as empty a slot in the speculative register file when the speculative data address stored therein does not become an architectural data address.

10. (Original) A digital signal processor as defined in claim 1, wherein the control logic is configured to update a control register corresponding to the one or more address parameters when a speculative data address is written to the speculative register file.

11. (Original) A digital signal processor as defined in claim 1, wherein the speculative register file comprises a circular buffer.

12. (Previously presented) A digital signal processor as defined in claim 1, wherein the speculative register file has more slots than the plural number of pipeline stages in the pipelined execution unit.

13. (Previously presented) A digital signal processor as defined in claim 1, wherein the speculative register file has two more slots than the plural number of stages in the pipelined execution unit.

14. (Currently amended) A method for operating a digital signal processor, comprising:

generating ~~a~~ speculative data addresses in response to address operands and one or more address parameters;

executing an instruction using data at a location specified by the speculative data addresses in a pipelined execution unit ~~having~~ including each of a plural number of pipeline stages required to execute the instruction from an initial stage in which the instruction is fetched through a final stage during which execution of the instruction is completed;

holding speculative data addresses in a speculative register file as a corresponding instruction advances through the pipeline, the speculative register file capable of storing a number of speculative data addresses equal to or greater than the plural number of pipeline stages in the pipelined execution unit and at least one architectural data address;

holding one or more speculative data addresses that have become architectural data addresses in the speculative register file; and

writing the speculative data address to the speculative register file as the speculative data address is generated by the address generator.

15. (Previously presented) A method as defined in claim 14, further comprising:

holding architectural data addresses in an architectural register file; and

moving an architectural data address from the speculative register file to the architectural register file in the event of a conflict for use of the speculative register file.

16. (Canceled).

17. (Original) A method as defined in claim 14, further comprising generating a next speculative data address based on a current speculative data address.

18. (Original) A method as defined in claim 14, further comprising marking as architectural an entry in the speculative register file when a corresponding instruction is completed by the pipelined execution unit.

19. (Original) A method as defined in claim 14, further comprising marking as empty a slot in the speculative register file containing an old architectural data address when a current architectural data address is defined.

20. (Original) A method as defined in claim 14, further comprising marking as empty a slot in the speculative register file when a speculative data address contained therein does not become an architectural data address.

21. (Original) A method as defined in claim 14, further comprising updating a control register corresponding to the one or more address parameters when the speculative data address is written to the speculative register file.

22. (Previously presented) The digital signal processor of claim 1, wherein the control logic is adapted to write an architectural data address to the architectural register file from the speculative register file only in the event that the speculative register file becomes full and requires a free location to write a new speculative data address.

23. (Previously presented) The method of claim 14 wherein the act of writing an architectural address from the speculative register file to the architectural data file is performed only in the event that the speculative register file becomes full and requires a free location to write a new speculative data address.